

21 December 2012

Guide to the NIEM 3.0 Release

Version 1.1

URI = <http://reference.niem.gov/niem/guidance/guide-to-niem3.0/1.1/>

This document was supported by Grant No. 2010-DB-BX-K041 awarded by the Bureau of Justice Assistance, Office of Justice Programs. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the U.S. Department of Justice.

Contents

1	INTRODUCTION.....	5
2	RELEASE PROCESS FOR NIEM 3.0	5
2.1	Schedule	5
2.2	Lead Domain Modeler (LDM)	6
2.3	Continuous preparation (always on-going).....	7
2.4	Domain Declaration of Intent to Input to NIEM 3.0.....	8
2.5	Pre-Alpha Stage (4.75 months).....	9
2.6	Review Teleconference	13
2.7	Alpha1 Stage (5 weeks).....	14
2.8	Alpha2 Stage (10 weeks).....	14
2.9	Beta1 Stage (8.5 weeks)	15
2.10	Release Candidate 1 (RC1) Stage (9 weeks)	16
2.11	NIEM 3.0 Operational Stage (4 weeks).....	17
3	NIEM HARMONIZATION PROCESS AND GUIDELINES	18
3.1	Harmonization Steps	18
3.2	Harmonization Difficulties	18
3.3	Data Sources for harmonization work.	19
3.4	Harmonization Stages.....	20
4	TOOLS AND COMMUNICATION FLOWS FOR NIEM 3.0	20
4.1	Tools and Facilities.....	20
4.2	NIEM 3.0 Input Formats	21
4.3	General NIEM Configuration Control Tool (NCCT) operations	21
4.4	How NIEM 3.0 will engage the public NIEM Community	23
5	NBAC STANDARD OPERATING PROCEDURES (SOP) FOR NCCT ISSUE PROCESSING	24
5.1	Initial issue pre-processing	24
5.2	NBAC meeting agenda.....	25

5.3	Before the NBAC meeting	26
5.4	NBAC meeting	26
5.5	Issue assignee responsibilities	27
5.6	NCCT attributes that support NBAC SOP.....	27
6	NCCT BASICS.....	28
6.1	NCCT Accounts	28
6.2	NCCT Mechanics	28
6.3	Recording the Resolution for an NCCT Issue	29
6.4	NCCT State Diagram	30
6.5	Sample NCCT issue page with metadata fields	31
7	NIEM DOMAIN UPDATE (DU) PROCESS.....	32
7.1	Domain evaluates the need to publish changes.....	33
7.2	Domain gathers requirements for the DU.....	33
7.3	Domain specifies changes in an approved input format.	33
7.4	Domain records each change represented by the DU in a NIEM XML change log.....	33
7.5	Iterative quality assurance (QA) process.	34
7.6	Release manager loads DU to staging server SSGT and generates DU schema.	34
7.7	Domain approves the DU schema and package.....	34
8	NIEM DATA MODELING GUIDE – QUESTIONS, CONSIDERATIONS, DECISIONS	34

1 Introduction

NIEM 3.0 will be executed in an accelerated, compressed release cycle, and therefore, will differ from previous release cycles in that it consists of constrained, partitioned, and carefully planned time periods. This guide outlines the NIEM 3.0 plan, details the steps in associated processes, identifies responsibilities, and provides guidance for executing.

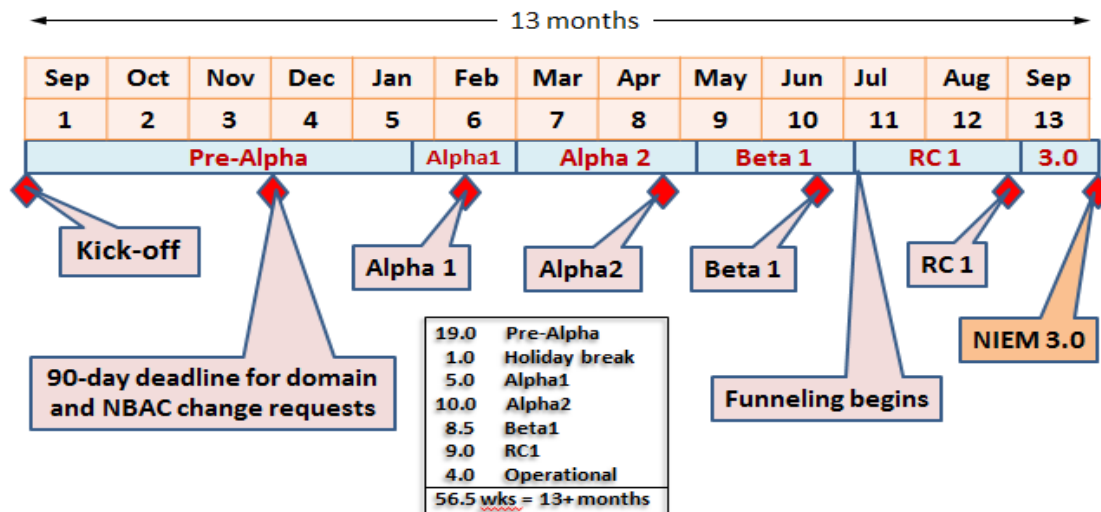
2 Release Process for NIEM 3.0

NIEM release stages defined below are in chronological order; however, the activities within each stage are not necessarily in chronological order because they are often iterative and concurrent. Although time constrained, this process strikes a balance between schedule driven deadlines and quality assurance. However, to meet tight deadlines in an compressed release cycle, domains and governance committees will have to coordinate and work efficiently during the Pre-Alpha stage (to be described below) to conduct analyses, harmonize, resolve issues, prepare changes and submit qualified content. Content submissions that do not qualify (pass format, conformance, and quality checks) or are late will not be integrated into 3.0.

Note that this accelerated process has no provision for stages beyond Alpha2, Beta1, and RC1. This process assumes that release 3.0 can be completed without the need for additional intermediate release stages. To ensure this, timelines are tight and deadlines are final. Additional pre-releases or extra time to meet deadlines will be allowed by explicit PMO approval only and will extend the release schedule.

2.1 Schedule

The following milestones are based on a 31 August 2012 start date. The milestones are somewhat tentative because they depend heavily on volume and quality of the changes input by domains and NBAC, as well as timeliness and the number of iterations required to reach consensus and achieve model stability. PMO has the option to extend timelines to accommodate missed deadlines and states of irreconcilable differences.



2.2 Lead Domain Modeler (LDM)

A *NIEM Domain* is a community of interest (COI) comprised of stakeholders that are aligned to the domain's mission space by virtue of affiliation, interest, or responsibilities.

The NIEM procedures require that each domain be sponsored by a Federal organization that has the ability to fund efforts to ensure a sustainable domain. The *Domain Steward* is required to sign a Domain Stewardship Agreement (DSA) which establishes a governing COI for a new dataset that will be added to NIEM in the form of a new domain, an extension to an existing domain, or modifications to the NIEM Core. Expectations of all parties with regard to the governance, oversight, and long-term maintenance are defined in the DSA.

Each NIEM domain is: (1) developed around mission area, (2) maintained by a self-sustaining governance body, referred to as a *Domain Steward* (or just a *Domain*), and (3) represented by a COI comprised of participants from one or many organizations, components, or agencies at the Federal, State, local or Tribal levels. Associated with the domain steward, the NIEM Domain Value Proposition Template defines the following persons:

Definition: Domain POC – The individual identified as the domain representative that will act as the liaison between the domain and the NIEM PMO and NIEM governance committees.

Definition: Domain Modeler – The individual(s) responsible for the development and maintenance of the domain model.

Definition: Domain Committee Support Staff - The individual(s) responsible for working with the Domain POC to manage the activity of the domain.

This guide will only define and refer to the *Lead Domain Modeler (LDM)*, who may or may not be the same person as the domain POC or a member of the domain committee support staff.

Definition: Lead Domain Modeler (LDM) – The person who is the domain’s single authoritative point of contact (POC) for content issues, development, and management. For a given domain, the LDM is the:

1. Primary representative who acts authoritatively on behalf of the domain for content issues.
2. Primary domain interface to the release manager and the other LDMs.
3. Domain content submitter (for major/minor release change requests and domain updates).
4. Person responsible for monitoring his/her domain content submission entry in NCCT (to be explained in **Section 2.4 Domain Declaration of Intent to Input to NIEM 3.0**).

The LDM may appoint a deputy if desired. The deputy will carry the same decision authority as the LDM in the event the primary LDM is unavailable for an immediate content decision.

When necessary during the 3.0 release cycle, and with approval from the appropriate LDM, the release manager may log NCCT entries for the LDM (e.g., if an issue or content submission comes to the release manager through an authorized alternate communication channel).

2.3 Continuous preparation (always on-going)

NIEM was designed to respond to changing requirements. The original and primary function of NBAC, NTAC, and domains is to regularly prepare plans for new and changing requirements. The more preparation for the next release that can be accomplished on a regular basis (i.e., publish domain updates, identify, design, and record new data content and architectural requirements, and resolve NCCT issues) the less resourcing impact the next release will have on committees and domains.

Before the kick-off date for NIEM 3.0, the following activities are considered preparation for a release cycle:

1. **Domain updates (DU)**: Domains publish DUs on their own schedules. Per the NIEM Version Architecture, these DUs are the basis for changes in future releases. Domains that maintain their content regularly through independent DUs are always setting up for the next major or minor release. This significantly spreads and reduces work that domains are otherwise forced to accomplish in a short timeframe immediately before or during a planned release. Instead of intensive analysis, compilation, and preparation of release change requests, their most current DUs are their release input. Similarly, domains can prepare change requests as soon as possible on their own timeline rather than waiting for a release kick-off. Either can prevent a last minute scramble to meet deadlines. (See **Section 7 NIEM Domain Update (DU) Process**.)

2. Issue resolution: The NIEM Configuration Control Tool (NCCT) records and tracks content and architectural issues collected from several outside sources including the Helpdesk, NBAC, NTAC, and LDMs. Committees and domains (should) meet regularly to review and resolve these issues. Resolutions are recorded for the next applicable release (major or minor). In this way, a significant portion of the changes for the next release are already decided, and the release manager will simply apply them in the next release cycle.
3. Harmonization: The harmonization master list (HML) identifies data components in NIEM that may (or may not) semantically overlap. Data components on the HML should be reviewed regularly by domains and committees. The more of these issues that can be dismissed or confirmed, if necessary resolved, and recorded in NCCT, the less work will be required in the next release. (Note that even dismissed harmonization issues should be recorded with rationale for dismissal to ensure the same issues are not raised a second time in the future.) The HML may be updated anytime a harmonization issue has been dismissed or resolved, and any time a new potential requirement to harmonize is identified. When developing new content, to reduce the need to add new items to the HML, NBAC and domains should always preview existing related NIEM data components for potential conflicts or duplication.

2.4 Domain Declaration of Intent to Input to NIEM 3.0

Release development involves a high degree of planning, and with so many participants (domains, NBAC, NTAC, etc.) there can be many unknowns. These include how many new data components and how many changes to existing components will be requested by domains; how much harmonization may be required after domains have developed content and changes; how difficult content issues or conflicts may be to resolve in governance meetings conducted after each review of an intermediate release product (Alpha1, Alpha2, Beta1, etc.); and how difficult it may be to integrate them. In an accelerated release, it is extremely important for all participants to see and understand the nature of ALL the content as soon as possible.

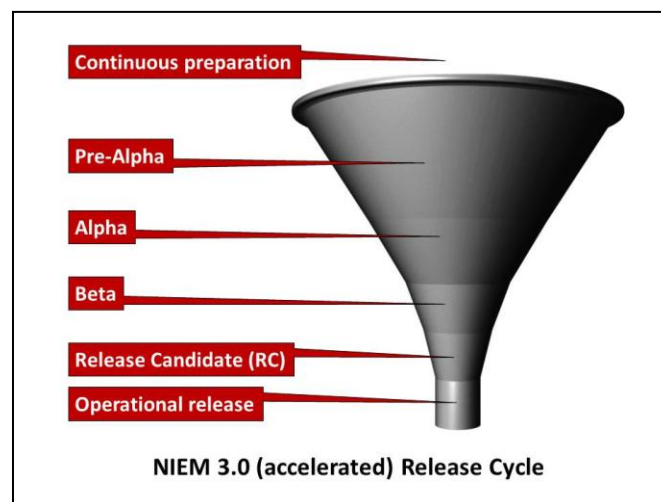
In order to encourage domain planning, reduce unknowns, and expose scope and information about domain inputs as early as possible, the NIEM 3.0 release process will require the following:

1. Not later than 14 September (two weeks after the kick-off), each domain must declare its intent to contribute content to NIEM 3.0.
2. Declaration must include:
 - a. Domain technical lead (name, domain, phone, email, organization).
 - b. Estimate date that first full draft input will be submitted.
 - c. What format? NIEM-conformant XSD? or change request spreadsheet (XLS)?
 - d. Estimated scope of final draft (i.e., number new and changed components).

3. Declare intent by one of two methods:
 - a. Open a new NCCT issue in the General Forum <http://tools.niem.gov/ncct/> (the preferred method).
 - b. Email to niem-comments@lists.gatech.edu (and the release manager will record NCCT issue).
4. A domain's NCCT declaration issue in the General Forum is visible to all NIEM participants (NBAC, NTAC, domains, PMO). The declaration will not be visible to the public.
5. Thereafter, all draft inputs, associated release manager feedback, and other correspondence will be recorded in that same NCCT issue for that domain only.
6. So, each domain will have a single NCCT issue in the General Forum that records files (the inputs), feedback, and correspondence associated with its domain input to NIEM 3.0.
7. Each LDM must have an NCCT account.

2.5 Pre-Alpha Stage (4.75 months)

The Pre-Alpha stage is the most dynamic period in the release cycle. It is characterized by preparation of new content and changes to existing content, harmonization, refactoring, integration, and application of architectural improvements. The entire release process can be viewed as a funnel where the Pre-Alpha stage is the top portion of that funnel.



1. Pre-Alpha stage begins on the assigned kick-off date (31 August 2012), and defines a period of activity that leads into the Alpha1 stage and the associated Alpha1 release of NIEM 3.0.
2. Concurrent activities conducted during Pre-Alpha (in no particular order):
 - a. Each domain identifies which (if any) of their domain updates should be applied to the new release (in NIEM 3.0 this only applies to CYFS domain – the only domain with an update in the publication area <http://publication.niem.gov/niem/domains/>).
 - b. Domains may also prepare content changes (additions, deletions, and modifications) in NIEM-conforming schema format or using the NIEM change request (download at <http://reference.niem.gov/niem/resource/change-request/1.0/>).

- c. Harmonization (See **Section 3 NIEM Harmonization Process and Guidelines** for details):
 - i. Each domain reviews the HML for data components in its namespace that appear to semantically overlap other components in Core, another domain, or its own domain namespace.
 - ii. Based on the HML, appropriate domains mutually determine if and how data components should be harmonized. Results are posted back into the HML (or a particular NCCT issue if not the HML).
 - iii. For a given release, the HML itself is one issue in NCCT.
 - iv. NBAC consults the HML for data components in NIEM Core that may require harmonization.
 - v. NBAC may also assist the domains with harmonization; particularly issues that involve NIEM Core.
 - vi. In the course of resolving harmonization issues, NBAC, a domain, or the release manager may identify and add new issues to the HML as needed.
- d. NTAC develops, discusses, and approves solutions for new requirements and architectural issues.
- e. As changes accumulate from NCCT and HML resolutions, DUs, and formal change requests, the release manager runs automated QA and conformance tests, inspects, applies simple corrections or adjustments where appropriate, and then integrates the changes. During this process, the release manager identifies additional content issues that may require further action by a domain, committee, or tiger team.
- f. Release manager integrates 3.0 architectural changes recommended by NTAC, and where appropriate, modifies content inputs to meet architectural changes.
- g. As soon as possible, release manager refers any issues and feedback resulting from inspection, testing, and integration to appropriate committees, domains, or tiger teams. This is recorded into the appropriate NCCT issue. This feedback:
 - i. Informs the domain of any minor corrections or adjustments to its inputs, and/or
 - ii. Requests additional action be taken to resolve, feedback, clarify, or justify portions of a change request or harmonization resolution.
- h. To be included in NIEM 3.0, all domain inputs (change requests) must qualify BEFORE the input deadline. To qualify, inputs must conform to the standard input format (NIEM schemas or CR spreadsheet), pass NIEM conformance, and pass quality checks (BEFORE the deadline).
- i. Domains must submit inputs early enough BEFORE the deadline to allow enough time to address significant issues identified in the review process.
- j. To assist domains, the release manager will accept domain content submissions in early draft form and provide feedback and guidance as soon as possible. This will

- help domains to identify content issues early when they are often quicker and easier to resolve.
3. All release inputs (including all domain change requests and DUs) must be *qualified*, and approved, and submitted NOT LATER THAN 3 calendar months (or 90 calendar days) after kick-off.
 - a. **The end of the 90-calendar-day input window is currently projected to be 30 November 2012.**
 - b. Domains should anticipate at least one or two submit/feedback/refine iterations to qualify content.
 - c. Domain change requests and DUs that do not qualify by after 90 days will NOT be integrated into the release. All such changes may be published by domains as DUs to NIEM 3.0 after 3.0 has been released, or in a subsequent major or minor release.
 - d. Domain and NBAC work on code list updates, harmonization, and unresolved NCCT issues may continue beyond 90 days (until the beginning of the Alpha1 stage). These resolutions and code updates will be integrated into the 3.0 release if schedule slack permits. Otherwise, resulting changes will be integrated into a later release or update. Core code list updates that could not be integrated, will be published as core updates after 3.0 has been released (the new 3.0 architecture will permit such).
 - e. On request, the release manager will review and QA check unfinished domain input. This can potentially identify systemic errors sooner to prevent wasted effort. Realize that the later this courtesy check is requested by a domain the more time it may take to receive feedback (since many other domains will be requesting checks or even submitting first or final draft inputs). The earlier a domain requests a check, the more likely it will receive feedback sooner and find potential systemic errors that could save time later.
 - f. **By 45 days, the release manager would like to see at least one draft input (which can be unfinished) from each registered domain.** This is not mandatory; however, since it will be logged into NCCT, it will assist everyone involved in NIEM 3.0 by exposing potential content early and possibly systemic errors (again, feedback will also be posted to NCCT within a domain's input issue).
 - g. Other work that will continue after the 90-calendar-day window includes:
 - i. Review and integrate change requests. This often identifies additional areas of harmonization to consider because domains make changes that collide or conflict without necessarily realizing it. Such issues may be sent to NBAC or appropriate domains to resolve. Obvious solutions may be applied by the release manager and provided to NBAC for review and approval.
 - ii. Integrate NTAC architectural improvements across the model. In particular, change requests submitted and qualified during the first 90 days will be in

- NIEM 2.1 form and therefore, the release manager must convert them to use the NIEM 3.0 architecture where applicable.
- iii. Research and integrate code list updates.
 - iv. As required, design optional extended representations for complex code lists.
 - v. Miscellaneous reviews of and cross-checks on changes by NBAC and domains as appropriate.
4. With PMO staff assistance, NBAC will research, identify, record, and provide all common (core) code list updates except FBI and domain codes. This includes both the code value updates as well as updated metadata for each code list. Identify and record metadata for all common code lists in NIEM Core (that will potentially require update). Metadata includes:
- a. Directory within NIEM release (where the schema resides).
 - b. Name or title of code list.
 - c. Short description of code list content and purpose.
 - d. Current authoritative source and POC.
 - e. Authoritative source document name.
 - f. Authoritative source Web site URL (where code list can be obtained).
 - g. For the base code list:
 - i. version#
 - ii. date (sometimes version# = date)
 - iii. URL where code list resides or was found
 - h. For code list updates/changes:
 - i. update# or change#
 - ii. date (sometimes change# = date)
 - iii. URL where code list updates/changes reside or were found
 - i. Publication formats (html, pdf, doc, docx, txt, etc.).
 - j. Additional notes or issues as necessary (e.g., classification of code list; if not on Web, describe how code list can be obtained, and how to identify and obtain updates in the future.)

The following are the metadata fields and one sample record (FBI) from the NIEM 2.1 code list spreadsheet:

NIEM code list metadata fields:

1. NIEM Release Subdirectory
2. Code Table Name
3. Description of Content
4. Authoritative Source
5. Source Publication Title
6. Version ID (or date)
7. Date
8. Update/Change number or date

9. File formats
10. URL to source publication base document
11. URL to source publication updates
12. Notes and Issues

Example record (FBI NCIC):

1. ncic_2000
2. National Crime Information Center (NCIC) 2000
3. Various NCIC code lists for reporting crimes
4. Federal Bureau of Investigation (FBI)
5. NCIC 2000 Code Manual
6. TOU 06-2
7. 2 Oct 2006
8. 06-2
9. pdf, doc
10. <http://www.leo.gov>
11. <http://www.leo.gov>
12. FOUO; now published as a database on secure www.leo.gov

2.6 Review Teleconference

Immediately before each intermediate release product review begins, the release manager will facilitate a teleconference with committees, domains, and other reviewers. The purpose of this meeting will be to refresh everyone regarding the process, explain the particulars of this release and its associated review, and respond to questions. After this teleconference, the release manager will publish the release package. This teleconference will cover:

1. What to expect in the release to be reviewed.
2. Feedback process, general roles, and specific assignments.
3. Deadline date(s) – NO feedback accepted after deadline.
4. What to review; how to review; what to look for; how to report it.
5. Where to send feedback (niem-comments@lists.gatech.edu or [NCCT](#)).
6. Answer questions as needed.
7. Valid feedback must contain (or may be rejected):
 - a. Submitter name, email, telephone, and organization.
 - b. If not a member of PMO staff, NTAC, NBAC, or a NIEM domain content development team (as is the case for any public comments), then a brief, but clear statement of experience or expertise with NIEM is required.
 - c. Complete, clear, actionable statements and associated recommendations.
 - d. Supporting examples as appropriate.

2.7 Alpha1 Stage (5 weeks)

Alpha stages focus on integrating the architectural improvements and the domain and core content changes submitted during Pre-Alpha. At this point, the release funnel narrows and no new large change requests will be accepted. Activity now centers on modifications that relate directly to the current set of data components. Adjustments, corrections, refactoring, harmonization, and integration operations continue and are coordinated through careful committee review, feedback, and approval of such actions.

1. Alpha1 starts 4³/₄ months after kick-off and after NBAC and domains have provided all qualified new and changed content, and NTAC has approved remaining architectural improvement designs.
2. On 11 January 2013 all content will be frozen and locked. This marks the end of the Pre-Release stage. At this point, no more refinements, code list updates, NCCT resolutions, or other changes will be accepted for integration into Alpha1. Additional changes that were not ready for Alpha1 may be applied to Alpha2 if and only if the schedule permits. Otherwise, they will be deferred to domain/core updates or future releases.
3. The Alpha1 stage begins the day after the Pre-Alpha stage closes.
4. In the first 4 weeks of the Alpha1 stage, the release manager:
 - a. Integrates remaining qualified domain inputs.
 - b. Integrates all remaining resolved NCCT issues.
 - c. Checks conformance and quality with the 3.0 architecture.
 - d. Generates Alpha1 release schemas, spreadsheet, change log, and QA report.
 - e. Posts Alpha1 products for review and testing by committees and domains.
 - f. Stages Alpha1 on NIEM tools staging server for usage testing.
5. Just before the review period, the release manager will facilitate a teleconference with reviewers (committees, domains, etc.). (See **Section 2.6 Review Teleconference.**)
6. Committees, staff, and domains review Alpha1 during the final 2 weeks of the Alpha1 stage, and provide feedback to the release manager by the end of that period.
7. Late feedback cannot be addressed in this stage, and will be deferred for action in the next stage (if still applicable).
8. Alpha1 will not be publicly available because at this stage the release will still be relatively dynamic, and review by governance committees and domains are expected to produce a significant number of issues and changes leading into Alpha2. Alpha2 and beyond will be more stable and available for review by the NIEM Community.

2.8 Alpha2 Stage (10 weeks)

Similar to Alpha1, the Alpha2 stage continues to apply changes needed to further stabilize the developing release.

1. In the first 2 weeks, the release manager reviews all feedback on Alpha1 and prepares a proposal that recommends actions to be applied to create Alpha2.
2. In a series of meetings over 2 weeks, NTAC, NBAC, and domains vet the feedback and recommendations for action, and reach consensus.
3. During the next 4 weeks, the release manager:
 - a. Integrates the approved resolutions/changes.
 - b. Time-permitting, integrates changes that were not ready for Alpha1.
 - c. Develops Alpha2 release schemas, spreadsheet, change log, and QA report.
 - d. Checks conformance and quality with the 3.0 architecture.
 - e. Posts Alpha2 products for review and testing by committees, domains, and the NIEM Community.
 - f. Stages Alpha2 on NIEM tools staging server for usage testing.
4. PMO publicly announces the NIEM 3.0 Alpha2 release on niem.gov:
 - a. Identifies URLs to the products and the tool staging server
 - b. Explains procedures for commenting.
 - c. Announces deadline for comments.
5. Just before the review period, the release manager will conduct a teleconference with reviewers (PMO, committees, domains, etc.). (See **Section 2.6 Review Teleconference.**)
6. In the final 2 weeks, committees, staff, domains, and the NIEM Community review Alpha2, and provide feedback to the release manager by the end of this stage.

2.9 Beta1 Stage (8.5 weeks)

The Beta stage begins when the developing release has stabilized to the degree that required changes are minor, and adjustments for architecture have been applied. The release schemas are in a state that will allow user beta testing. Changes are still allowed; however, these are much smaller in magnitude and complexity and are generally only corrections and adjustments. Examples of Beta stage changes include adjustments to data component structure, movement of properties; renaming and redefining; addition or refinement of keyword, usage, and example data (associated with definitions); addition of a very small number new components (essentially inadvertent omissions or additions needed to adjust structure), and occasionally very minor corrections to architecture. The NIEM 3.0 process simply assumes after applying the feedback received on Alpha2 the developing release will be characterized as described above (Beta).

Note also that a normal Beta stage should allow several months for beta testing. However, under the shortened NIEM 3.0 release cycle there is essentially no time for such.

1. In the first 1½ weeks, the release manager reviews all feedback on Alpha2 and prepares a proposal that recommends actions to be applied to create Beta1.
2. In a series of meetings over the next 1½ weeks, NTAC, NBAC, and domains vet the feedback and associated recommendations for action, and reach consensus.

3. In the next 3½ weeks, the release manager:
 - a. Integrates the approved solutions/changes.
 - b. Time-permitting, integrates changes that were not ready for Alpha2.
 - c. Develops Beta1 release schemas, spreadsheet, change log, QA report, alternate database formats.
 - d. Checks conformance and quality with the 3.0 architecture.
 - e. Posts Beta1 products for review and testing by committees, domains, and the NIEM Community.
 - f. Stages Beta1 on NIEM tools staging server for usage testing.
4. PMO publicly announces the NIEM 3.0 Beta1 release on niem.gov:
 - a. Identifies URLs to the products and the tool staging server
 - b. Explains procedures and deadline for commenting.
5. Just before the review period, the release manager will conduct a teleconference with reviewers (committees, domains, etc.). (See **Section 2.6 Review Teleconference.**)
6. During the final 2 weeks, committees, staff, domains, and the NIEM Community review Beta1, and provide feedback to the release manager by the end of this stage.

2.10 Release Candidate 1 (RC1) Stage (9 weeks)

The Release Candidate (RC) stage marks the period in which the developing release is essentially completed. An RC should be identical to, and therefore stable enough to become the final operational release. Only extremely minor corrections to the release are allowed. These include minor corrections to character strings, names, definitions, namespaces, etc. At this stage absolutely NO architectural or significant content changes are allowed. Any such changes will require regeneration of the release as a Beta product again. The RC stage can be deceptively easy because the release is stable and another review seems repetitious; reviewers can easily become apathetic. However, this is the final review and diligence must ensure quality *before* publication.

1. In the first 2 weeks, the release manager reviews all feedback on Beta1 and prepares a proposal that recommends actions to be applied to create RC1.
2. In a series of meetings over the next 2 weeks, NTAC, NBAC, and domains vet the feedback and associated recommendations for action, and reach consensus.
3. In the next 4 weeks, the release manager:
 - a. Applies approved minor changes.
 - b. Develops RC1 release schemas, spreadsheet, change log, QA report, MPD artifacts, alternate database formats, and SSGT, ConTesA, and Code XSD Generator.
 - c. Checks conformance and quality with the 3.0 architecture.
 - d. Posts RC1 products for final review and testing by committees, domains, and the NIEM Community.

- e. Stages RC1 on NIEM tools staging server for final usage testing.
- 4. PMO staff prepares announcements and public relations materials for niem.gov.
- 5. PMO announces the NIEM 3.0 RC1 release on niem.gov:
 - a. Identifies URLs to the products and the tool staging server.
 - b. Explains procedures and deadline for commenting.
- 6. Immediately before the review period, the release manager will conduct a teleconference with reviewers (committees, domains, etc.). (See **Section 2.6 Review Teleconference.**)
- 7. During the final (1) week, committees, staff, domains, and the NIEM Community review RC1, and provide feedback to the release manager by the end of this stage.

2.11 NIEM 3.0 Operational Stage (4 weeks)

- 1. In the first (1) week, the release manager reviews any feedback on the RC and prepares a proposal that recommends actions to be applied to create the final operational NIEM 3.0.
- 2. During the next (1) week, NTAC, NBAC, and domains vet and agree on the recommended refinement actions.
 - a. If RC1 receives no valid feedback to necessitate further refinement, it will be the basis for the final operational release.
 - b. Otherwise, if refinements are necessary to RC1, the committees will recommend to the PMO whether:
 - i. Changes should be made directly within the final release (for extremely minor changes only), or
 - ii. RC2 is required (which could extend the timelines), or
 - iii. Beta2 should be generated (for extensive or serious issues; and would extend the timeline), or
 - iv. Feedback should be addressed after NIEM 3.0 is released (i.e., in a future release, domain update, or core update).
- 3. In the final 2 weeks, the release manager will:
 - c. Apply final approved refinements.
 - d. Convert the RC to a final operational NIEM 3.0 release by removing "RCn" from version attributes of all release products.
 - e. Run final checks for conformance and quality with the 3.0 architecture.
 - f. Post NIEM 3.0 products to release.niem.gov and reference.niem.gov.
 - g. Move NIEM 3.0 into the NIEM tool production server at tools.niem.gov.
- 4. PMO conducts final review of release products.
- 5. PMO posts announcement of the NIEM 3.0 operational release on niem.gov with associated URLs to products and tools.

3 NIEM Harmonization Process and Guidelines

In NIEM, harmonization is the process of resolving semantic overlap and duplication among data representations. Due to the large size of the model and the variety of requirements, sources, and contributors, some degree of overlap is to be expected. Therefore, harmonization is a standard, explicit part of the release building process that controls and minimizes this potential duplication. Sources of overlap may include:

- a. Contributor is unaware of an existing component and adds a semantically identical or similar component.
- b. Two contributors are unaware of each other's changes and each adds a new, semantically identical (or very similar) component to a release.
- c. Contributor finds a component from the model that almost matches requirements, but can't be used as is and the ability to make the necessary adjustments is constrained (e.g., frozen namespaces in a minor release, etc.).

3.1 Harmonization Steps

- a. Identify potential locations of overlap or duplication in the model.
- b. Evaluate.
 - i. Determine if overlap or duplication actually exists.
 - ii. Identify any additional requirements; for example, generalize component for a broader audience, move to Core, etc.
- c. Resolve overlap.
 - i. Remove duplication.
 - Remove duplicate.
 - Replace with reference (if needed) to remaining component.
 - ii. Merge.
 - Refactor as new component in a domain or Core.
 - Replace originals with references (if needed) to new component.
 - iii. Refactor as proper NIEM construct. Create appropriate substitution group head/member(s), role, augmentation, association, etc.
 - iv. Other, as determined by situation.
- d. Resolve non-overlap. Ensure component names and/or definitions are clear, meaningful, and distinct from each other.

3.2 Harmonization Difficulties

- a. Introducing and/or identifying semantic overlap.
 - i. Practically impossible to know everything in model.

- ii. Search effectiveness tends to be hit or miss, and therefore, inefficient.
- iii. Domain changes are applied once they pass QA; little or no period of socialization or peer review.
- iv. Overlapping components often have different names and can be located in distinctly different subject areas of the model, making detection difficult.
- v. Comprehensive review of the model to look for overlap is a very tedious and difficult process.
- vi. Can be difficult to identify and review some content without subject matter expertise.
- b. Tool support is limited; harmonization is primarily based on semantics; requires manual comparison and review.
- c. Even when overlap is known, it is not always possible to resolve immediately.
 - i. Locked Core and Core dependencies in minor releases can limit harmonization efforts.
 - ii. Specific domain requirements can sometimes limit available solutions.
- d. Domain or cross-domain collaboration is often required.
- e. Can be difficult to harmonize different requirements into a single resolution that works for all involved parties.

3.3 Data Sources for harmonization work.

- a. NIEM Releases.
 - i. Core and dependencies
 - ii. Domains
 - iii. Code tables
- b. Domain Updates.
 - i. All domain updates must be harmonized back into the originating domain for the following release. Much of this work will be different than the standard harmonization process as overlap is expected and a domain update change log should be provided, identifying overlap and how it should be resolved.
 - ii. Standard harmonization may still be needed for new components introduced in a domain update and for components not properly labeled in the change log.
- c. Information Exchange Package Documentation (IEPDs).
- d. NCCT harmonization issues and the harmonization master list (HML).
- e. Other known issues (that should be in NCCT).

3.4 Harmonization Stages

There are essentially 3 stages of harmonization:

- a. Input
 - i. Preventative: Perform harmonization on input prior to updating the model.
 - ii. Prevents the initial introduction of overlap.
 - iii. Use automated testing and improved content awareness and domain communication to allow submitters to resolve their own harmonization issues while preparing inputs and changes.
- b. Pre-release
 - i. Overlap introduced into the model but resolved before final release.
 - ii. Perform harmonization checks as part of testing - both automated and targeted searches.
 - iii. Even though identified, some of these may not be resolvable (e.g., during a minor release cycle).
- c. Post-release
 - i. Identified via bug reports, additional testing, or may have been identified but unresolvable in this release.
 - ii. Capture feedback identifying overlap (NCCT, IEPD developer, etc.).
 - iii. Begin harmonization efforts for next release.

4 Tools and Communication Flows for NIEM 3.0

NIEM 3.0 will consist of constrained and carefully partitioned time periods. The plan for NIEM 3.0 incorporates various tools to support the detailed committee and domain work to harmonize, resolve issues, review, and feedback. This whitepaper describes tools, communication flows, and the processes for accomplishing this work.

Except for timelines and input qualifications, neither the NIEM 3.0 plan nor this paper impose constraints on how committees, domains, and staff approach, organize, meet, prioritize, or accomplish the work necessary to produce content for NIEM 3.0.

4.1 Tools and Facilities

The following facilities / tools will be used during NIEM 3.0 development:

- a. NIEM Configuration Control Tool (NCCT) – <http://tools.niem.gov/ncct/>
- b. NIEM Comments Mailing List – niem-comments@lists.gatech.edu
- c. National Information Sharing Standards Helpdesk - NISShelp@ijis.org and <http://www.it.ojp.gov/framesets/niss-noClose.htm>

4.2 NIEM 3.0 Input Formats

New releases address improvements, corrections, and new and changing requirements. There are three general classes of inputs into NIEM 3.0:

- a. New content (that does not already exist in NIEM).
- b. Changes to existing content.
- c. Issue resolutions (includes harmonization solutions).

These classes are not necessarily distinct. For example, an issue resolution usually results in either new content or changes to existing content. However, an issue resolution tends to be limited in its scope of impact, usually limited to a particular data component or a couple of components. On the other hand, changes to existing content requested by a domain are often quite extensive.

Therefore, there are four general formats for submitting inputs into NIEM 3.0:

- a. Domain update (DU) – A formal Model Package Description (MPD) used for large scoped changes from domains. DUs usually exist before a release cycle begins, and at the request of the senior domain technical POC, will be integrated into the release. Unless it already exists, preparation of a DU as input to 3.0 is discouraged because of the highly constrained timeline. Other formats have fewer requirements to satisfy. For DU requirements, consult the Domain Update Specification <http://reference.niem.gov/niem/specification/domain-update/1.0/>
- b. NIEM-conforming schema (XSD) – used for relatively large scoped change or new content; formal; requires knowledge or understanding of the NIEM NDR.
- c. Change Request – <http://reference.niem.gov/niem/resource/change-request/1.0/> – A formally structured spreadsheet used for relatively large scoped changes or new content; does NOT require detailed knowledge or understanding of the NIEM NDR.
- d. Informal written description of changes or additions associated with the resolution of an NCCT issue or harmonization; used for changes that are relatively small in scope and easy to describe in text.

4.3 General NIEM Configuration Control Tool (NCCT) operations

The NIEM Configuration Control Tool (NCCT) is an issue tracking tool used to record, track, and manage issues, including error reports and new requirements, for NIEM releases and pre-releases. NCCT contains forums for the various governance committees and domains. Members of the NBAC, NTAC, LDMs and their development team members, PMO staff, and the National Information Sharing Standards (NISS) Help Desk are all eligible for accounts, allowing them to directly submit new issues and comment on existing ones.

The public NIEM community does not have access to NCCT. However, for NIEM 3.0 the public can email or call NISS Helpdesk, which can submit NCCT issues on their behalf. They can also email an issue to niem-comments@lists.gatech.edu and the release manager will add it to NCCT. A new public NCCT forum (NIEM-3.0) has been established for this purpose.

The committee or domain that owns a forum is responsible to resolve issues in that forum. See **Section 5 NBAC Standard Operating Procedures (SOP) for NCCT Issue Processing** for more details about how committees and domains resolve NCCT issues. Any member of a committee or domain who has an NCCT account and correct WRITE permissions for a forum may do the following to issues in that forum:

- a. Create a new NCCT issue.
- b. Enter a comment or reply to a comment.
- c. Update metadata attribute fields (ensure you know what they mean).
- d. Attach (upload) supporting files.
- e. Change the state of an issue.
- f. Add their email address to be cc'd on any future activity (changes) on an issue.
- g. Assign an issue to himself/herself as lead.

Issue resolution may require off-line discussions by members through comments and replies in the issue fields. There is no limit to the number or size of comments or replies.

NCCT issues move through various states from OPEN to CLOSED associated with their status:

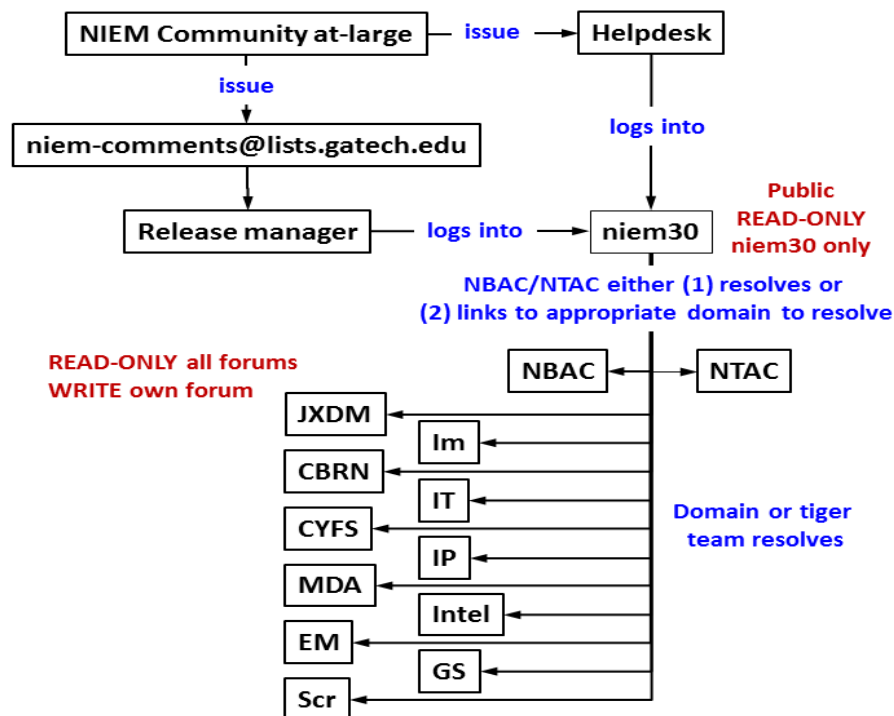
1. Open issues. An issue whose state is “UNCONFIRMED”, “NEW”, “ASSIGNED”, or “REOPENED” is considered an unresolved, open issue. The objective of each committee and domain is to change the state of each open issue to “RESOLVED”.
2. Duplicate issues. The state of an issue determined to be a duplicate of another (whether resolved or unresolved) should be “RESOLVED / DUPLICATE”. To do this, check “Resolve NCCT issue, mark it as duplicate of NCCT issue # ___”, enter the issue ID it duplicates in the box, and commit the state change by clicking the “Commit” button.
3. If consensus is reached on a resolution for a given issue, then some member of the committee/domain should:
 - a. Record a final comment that clearly and completely states exactly how the issue is to be resolved within NIEM 3.0. This statement should also include a rationale if appropriate.
 - b. Attach and reference supporting files if needed.
 - c. Revise the state of the issue to “RESOLVED / FIXED”.
4. If the committee/domain determines that a given issue is invalid or a change to the model is unnecessary, then some member of the committee/domain should:
 - a. Record a final comment that states the decision and the rationale.
 - b. Revise its state to “RESOLVED / INVALID”, or “RESOLVED / WONTFIX” (which ever is most appropriate).
 - c. Do not use “WORKSFORME” or “MOVED”. These resolved states are generally not applicable to content issues.
 - d. After confirming the state change, the committee/domain member should change the state (again) to “CLOSED”.

5. For each “RESOLVED / FIXED” issue, the release manager will integrate the resolution into NIEM 3.0 and change its state to “VERIFIED”.
6. For each “VERIFIED” issue, when a final release containing this resolution has been published, the release manager will change its state to “CLOSED”.

4.4 How NIEM 3.0 will engage the public NIEM Community

1. The NIEM Community (or the public) may submit comments (issues, suggestions, corrections, recommendations, etc.) into the NIEM 3.0 release process at any time during the release cycle.
2. A new publicly readable NCCT forum (labeled “NIEM-3.0”) will track public comments. To read the issues and responses in this forum, no account or permissions are necessary. Point a browser to <http://tools.niem.gov/ncct/> and click the “search” link. All searches executed outside of an account are constrained to the NIEM-3.0 forum.
3. There are two ways for the public to comment:
 - a. NISS Help Desk: email NISShelp@ijis.org (or call 877-333-5111 or 703-726-1919)
 - b. Send email to niem-comments@lists.gatech.edu
4. Valid feedback must contain:
 - a. Submitter name, email, telephone, and organization.
 - b. Clear statement of experience or expertise with NIEM (unless the feedback corrects simple or obvious errors; e.g., typographical, spelling, grammar, punctuation, etc.).
 - c. Clear, complete, actionable statements and associated recommendations.
 - d. Supporting examples as appropriate.
5. Both the Helpdesk and the release manager can log issues into the NIEM-3.0 forum. For each issue received through NISS Help Desk or niem-comments, a new NCCT issue will be created in NIEM-3.0 and assigned a unique integer identifier that contains the original text of the email.
6. The submitter will receive a response from the Helpdesk or release manager that provides the ID and URL for the new NCCT issue, so that he/she can track progress. Example: http://tools.niem.gov/ncct/show_bug.cgi?id=421
7. If later the submitter has additional information, examples, etc., or if NBAC/NTAC requests such, he/she can email it and refer to the issue ID. That email will be logged as an additional comment under its issue ID in NIEM-3.0 (in example above, issue #421).
8. The intent of this mechanism is to (1) collect feedback and comments from the field without allowing those who may not be familiar with NIEM governance processes to cause unnecessary delay by grandstanding or perpetuating marathon discussions with committees; and (2) to spare the public from viewing “sausage-making” aspects of NIEM governance.
9. As part of its business, NBAC and NTAC will
 - a. Review appropriate issues in the NIEM-3.0 forum and determine if they are valid and can be resolved quickly.

- b. If invalid or cannot be addressed, a rationale will be recorded in the NIEM-3.0 forum.
- c. If a resolution or response can be determined quickly, it will be entered in NIEM-3.0. Otherwise, the issue will be assigned to an appropriate governance group to resolve.
- d. Issues that require more work will be linked to a new issue in the appropriate forum for action in accordance with that group's SOP for issue resolution.
- e. Once resolved, the resolution and rationale will be recorded in the NIEM-3.0 forum so the submitter and others can see how it was handled.



Communications flow for public comments.

5 NBAC Standard Operating Procedures (SOP) for NCCT Issue Processing

After issues have been entered into appropriate NCCT forums by account holders, the following sections outline the NBAC SOP for resolving these issues; however, other governance groups apply similar procedures (NTAC, JXDM domain, etc.).

5.1 Initial issue pre-processing

The release manager initially reviews, assesses, and pre-processes all NCCT issues as follows:

- a. Filter and move issues to appropriate forum.
 - i. If an issue is already in the appropriate forum (NTAC, NBAC, domain, etc.), then it should remain there.
 - ii. NIEM 3.0 issues received from the public (through Helpdesk or niem-comments email) will always be recorded directly into the NIEM-3.0 forum. These will be linked to the appropriate committee or domain forum for resolution.
 - iii. The General forum is a collection point for issues that cannot be recorded by a submitter in the appropriate forum. The release manager will move issues recorded in the General forum to the appropriate forum.
 - iv. In general, content issues are moved to NBAC forum, technical or architectural issues to NTAC, Justice-specific issues to JXDM, etc.
- b. Set target release; this attribute identifies the next release for which it is possible to implement the resolution of the issue.
 - i. Core or Core-dependent issues target the next major release.
 - ii. Domain and other non-Core dependent issues target the next major or minor release.
 - iii. Issues that are not resolved in time for their target release are retargeted for a future release.
- c. Set additional metadata on the issue as needed, including the 'Category' field (values are: *Harmonization, New content, Modify data, Long term*, etc.).
- d. Comment on and close an issue that is obviously invalid. An issue may be invalid for one of several reasons, including (but not limited to):
 - i. Duplicates a previous issue (will be linked to the issue it duplicates).
 - ii. Already overtaken by events (record rationale).
 - iii. Falls outside the scope of NIEM (record rationale).
- e. Comment on and resolve an issue if the solution is obvious and does not require NBAC or SME discussion. For example:
 - i. Misspelling in a name or definition.
 - ii. Minor error in a definition.
 - iii. New or updated code values from authoritative source for a code list.
- f. Propose issues to be triaged and issue resolutions to be reviewed at next NBAC engagement (NBAC can reprioritize).
- g. Assess difficulty.
- h. When appropriate, recommend persons who are not NBAC members that should be present for NBAC discussion (e.g., a particular SME, the submitter, etc.).
- i. Prepare supporting information as needed for next meeting agenda and email to NBAC coordinator.

5.2 NBAC meeting agenda

- a. NBAC Coordinator prepares and emails agenda at least 3 workdays before the NBAC meeting.
- b. Agenda will contain next issues to triage/assign and issues with proposals for review.
- c. Interested participants can volunteer at the next meeting or via email to NBAC coordinator.

- d. Each NCCT issue listed in agenda will contain (as appropriate):
 - i. Issue ID linked directly to details in NCCT. Example: As long as account holder is logged into NCCT, then issue ID="149" can be viewed with URL syntax:
http://tools.niem.gov/ncct/show_bug.cgi?id=149
 - ii. Short name/description of the issue.
 - iii. For issues with proposed resolutions: assignee (individual or team leader).
 - iv. Only if needed, for issues with proposed resolutions: short statement of status.

5.3 Before the NBAC meeting

Each NBAC member reviews agenda to:

- a. Determine if any issues to be triaged are of interest (volunteer via email to NBAC coordinator if unable to attend meeting).
- b. Evaluate proposed resolutions recorded in NCCT.
- c. Prepare an alternative for any proposed resolution believed to be unacceptable.
- d. Time is critical, so SILENCE IS ACCEPTANCE!

5.4 NBAC meeting

NBAC meets for NCCT issues every Thursday 2-3pm ET unless otherwise scheduled. This meeting is generally partitioned into 2 parts:

- a. Part 1: Triage and assignment of each new issue:
 - i. Co-chairs take interested volunteers and/or make assignments based on expertise.
 - ii. Release manager or NBAC Coordinator records assigned individual or team leader as NCCT "assignee" field.
 - iii. Release manager or NBAC Coordinator records team members in NCCT "cc:" field.
 - iv. As appropriate, co-chairs issue instructions and suspense date to assignee.
 - v. If it is determined that an issue duplicates another, it will be marked as such in NCCT and resolved together as part of its duplicate.
- b. Part 2: Review proposed resolutions:
 - i. Co-chairs open the floor.
 - ii. Issue team leader may present very brief summary of issue resolution proposed.
 - iii. Short discussion and questions (rule of thumb: 5-10 minutes maximum time).
 - iv. Co-chairs close floor for vote, alternative proposals, or deferral for more work.

Approvals and disapprovals with rationale are formally recorded in NCCT. As appropriate, all outcomes are reported back to the source with resolution and rationale. Approved resolutions are implemented in the next possible release or update. Disapprovals are closed without action, except to record rationales in NCCT.

5.5 Issue assignee responsibilities

- a. Organize tiger team meetings as needed to resolve an issue and recommend resolution.
- b. Record proposed tiger team resolution directly into NCCT prior to agenda publication for consideration at next meeting.
- c. Ensure NCCT record contains details and any appropriate attachments to implement.
- d. Attend the NBAC meeting at which his/her proposed resolution(s) will be reviewed.
- e. Keep solutions as simple as possible and try to avoid large impacts to the model without good reason. It is usually better to close a difficult issue without action, rather than design a complex, impractical solution for a rare use case.
- f. Ensure that each team member understands that he/she represents a large number of people and organizations. Therefore, it is an important responsibility to consider the impacts of an issue resolution on other domains and the NIEM user community at large (in addition to one's own local requirements).

5.6 NCCT attributes that support NBAC SOP

The NBAC SOP uses several NCCT attributes to track issues:

Assignee: Person or tiger team leader to whom issue has been delegated.

<i>Process state</i> (values):	Where an issue is within the NBAC process.
New	Awaiting initial preparation for NBAC.
TBA - to be assigned	Prepared for triage and assignment in NBAC.
WKG - in progress	Unassigned and in progress.
ASN - assigned	Assigned and in progress.
APP - tentatively approved	Resolution approved pending 1-week waiting period for members who were not in attendance.
REC-recommendation posted	A proposed recommendation is ready to discuss.
RES - resolved	Resolution has been recorded/approved by NBAC.
Other	(Special case used when all other values are N/A.)
<i>Category</i> (values):	General nature or type of an issue.
Domain change request	Issue with attached spreadsheet or XSD containing a request for significant domain modifications and/or new content.
General	(Special case used when all other values are N/A.)
Harmonization	Self-evident
Long Term	Issue deferred for later action.
Modify data	Self-evident
New content	Self-evident

Target (release or milestone): Release version to which resolution would be applied.

6 NCCT Basics

This is a very basic introduction to the NIEM Configuration Control Tool (NCCT), an adaptation of the open source tool, [Bugzilla](#). For more details that explain how to use NCCT in NIEM see the [NCCT User Guide](#). (<http://reference.niem.gov/niem/guidance/ncct-user-guide/0.9/>)

6.1 [NCCT Accounts](#)

1. Each committee and domain member must have an NCCT account with the appropriate write permissions in order to actively participate and write comments within a forum.
2. New accounts can be obtained from: <http://tools.niem.gov/ncct/createaccount.cgi>
3. Once you have an account, you must send email to pgmw-system@gtri.gatech.edu and request permission to read all NCCT and write to the appropriate forum(s).
 - a. Request password resets through: pgmw-system@gtri.gatech.edu
 - b. See the [NCCT User Guide](#) for a list of NCCT metadata fields, code values, and definitions. Some of those fields and definitions are listed below in the next section.
 - c. Each issue has a unique integer identifier assigned at creation time. While logged into NCCT with a valid account login/password, a user can view an issue using its URL. Example, URL for issue 400 is http://tools.niem.gov/ncct/show_bug.cgi?id=400.
 - d. In an Excel spreadsheet, if cell B2 contains an NCCT issue integer identifier then a formula for a cell containing a hyperlink directly to the issue page for that identifier is:
`=HYPERLINK(CONCATENATE("http://tools.niem.gov/ncct/show_bug.cgi?id=",B2),B2)`
4. NCCT permissions are as follows:
 - a. NIEM-3.0 (a relatively new forum set up for NIEM 3.0) is the only forum that allows READ-ONLY access to the public.
 - b. Only the NISS Helpdesk and the release manager can WRITE issues to NIEM-3.0.
 - c. Each NCCT account holder has READ-ONLY access to ALL forums (so that each committee and domain can see across NCCT).
 - d. Each member of a committee or domain can WRITE to the corresponding forum he/she is a member of (but no others).
 - e. If necessary, permissions can be changed to accommodate WRITE access and discussion across domains and committees.

6.2 [NCCT Mechanics](#)

1. All NCCT issues are assigned consecutive integers based on creation date. A new issue is always assigned an ID equal to the next consecutive unused integer in NCCT.
2. NCCT issues can be easily cross-linked (hyperlinked) to other issues through their IDs.
3. Support files may be attached to any NCCT issue page.
4. NCCT issue status options:
 - a. UNCONFIRMED – An issue that has been added to NCCT but has not yet been confirmed as a valid issue.
 - b. NEW – An issue that has been confirmed to be valid, but not yet assigned or resolved.

- c. ASSIGNED – Identifies the issue lead person assigned to work on and propose recommendations for resolution.
- d. REOPENED – An issue that was previously closed but must be addressed again.
- e. RESOLVED – An issue that is resolved but has yet to be implemented in a release.
- f. VERIFIED – An issue that has been implemented but not yet published in a release.
- g. CLOSED – An issue that
 - i. Has been resolved and **requires no further action**.
 - ii. Release manager will close issues requiring content changes only after those changes have been published in a release.
 - iii. An issue determined to be invalid and/or requires no action (rationale required).

5. NCCT RESOLVED issue options:

- a. FIXED – A solution for the issue has been approved and documented.
- b. INVALID – The issue is determined to be outside scope of NIEM, incorrect, or otherwise invalid.
- c. WONTFIX – The issue is determined to be valid but will not be resolved.
- d. DUPLICATE – The issue is a duplicate of another issue in the NCCT (and is cross-linked to it).
- e. WORKSFORME – The issue cannot be recreated (needs more information or is invalid).
- f. MOVED – The issue will be managed and addressed outside of NCCT.

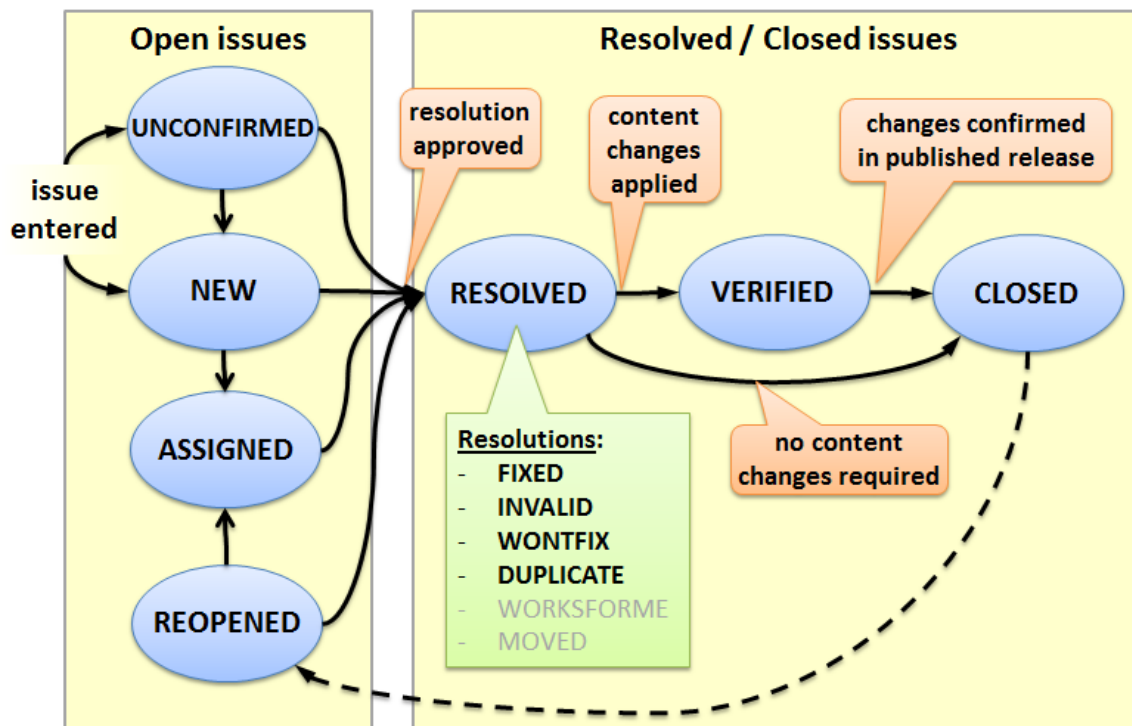
6.3 Recording the Resolution for an NCCT Issue

Once an issue has been resolved and its resolution approved, one NBAC member is assigned to enter the resolution as a final comment in NCCT.

1. For issues that will NOT result in content changes to the data model, assignee enters final comment, marks issue as 'Resolved', and then closes issue.
2. For issues that will result in content changes to the data model, assignee enters final comment only.
 - a. Include enough detail to implement the solution, including NDR-conformant names, definitions, data types, code values and code value definitions, etc. as applicable.
 - b. For very small updates to model (e.g., a new property or two), a textual description of the change is sufficient.
 - c. For non-trivial updates to model, attach a change request file, a specially formatted spreadsheet or XML input file that allows automated processing of the data. Change request available here: <http://reference.niem.gov/niem/resource/change-request/1.0/>
 - d. Attach any support files needed to understand or execute resolution.
3. Release manager processes content-changing issues.
 - a. Marks issue as 'Resolved' once it has confirmed there is sufficient detail to implement.
 - b. Marks issue as 'Verified' once it has implemented the content changes.
 - c. Marks issue as 'Closed' only AFTER changes have been implemented in the model and published in a release.

6.4 NCCT State Diagram

The following diagram indicates how NCCT issues change state:



6.5 Sample NCCT issue page with metadata fields

The following sequence of illustrations is a sample NCCT issue record indicating metadata, descriptions, and discussions recorded for each NIEM issue.

NCCT Issue#: 89 Process state: Other Reporter: test@test.org

Forum: NBAC Issue Type: Add CC: CC:

Category: New content Version: 1.0 Priority: P2 Severity: normal Target Release: NIEM 3.0

Status: NEW **Current status**

Resolution: Assigned To: default@ncct.gov URL: Summary: Additional information needs to be added to PassportType. **Metadata attributes (to classify)**

Summary; repeat in issue description below (because this can change!)

Attachment	Type	Created	Size	Actions
Details of proposed change	application/octet-stream	2012-02-09 09:02	19.37 KB	Edit
Create a New Attachment (proposed patch, testcase, etc.)				View All

Attached files

NCCT Issue 89 depends on: [Show dependency tree](#)

NCCT Issue 89 blocks: [Show dependency graph](#)

Additional Comments: ☐ Private **Enter comment**

cc: to your email

☐ Add test@test.org to CC list **Sample NCCT Issue ...**

Only users in all of the selected groups can view this NCCT issue:
(Unchecking all boxes makes this a more public NCCT issue.)

☐ GTRI

Users in the roles selected below can always view this NCCT issue:
(The assignee can always see a NCCT issue, and this section does not take effect unless the NCCT issue is restricted to at least one group.)

☒ Reporter ☒ CC List

Used to set/reset current status

☒ Leave as NEW

☐ Accept NCCT issue (change status to ASSIGNED)

☐ Resolve NCCT issue, changing resolution to FIXED

☐ Resolve NCCT issue, mark it as duplicate of NCCT issue #

☐ Reassign NCCT issue to default@ncct.org

☐ Reassign NCCT issue to default assignee of selected component

Commit (this record)

[View NCCT Issue Activity](#) | [Format For Printing](#) | [Clone This NCCT Issue](#)

Description of issue (cannot be changed)

Description: [\[reply\]](#) **Opened:** 2006-01-10 12:00

☐ Private

Additional data about Passports exists and should be included in the PassportType component.

Originator: BISED Team

Namespaces: InformationAssurance

Components: PassportType

Recommendation: Suggest including the following properties to the PassportType component:

Name on Passport, Country of Issuance, City of Issuance, State/Province of Issuance, Location of Visa Application, Visa Application Date, Visa Application Disposition, Previous Passport Name.

Comment #1 (Move to NBAC)

----- Comment #1 From Mark Kindl 2008-06-05 12:41 [reply] ----- ☐ Private

Move to NBAC

----- Comment #2 From [Christina Medlin](#) 2009-01-22 14:04 [reply] -----

Changed target release to 3.0.

----- Comment #3 From [Deborah Brooks](#) 2012-02-09 09:02 [reply] -----

[Created an attachment \(id=101\)](#) [edit]
Details of proposed change

----- Comment #4 From [Deborah Brooks](#) 2012-02-09 09:03 [reply] -----

RECOMMENDED RESOLUTION #9
Note: This recommendation would necessitate a change to NIEM Core for consistency and completeness.

1. Since a passport is really a specialization of a document, redefine the existing nc:PassportType as an extension of nc:DocumentType, adding whatever attributes this specialization requires. See scr:PassportAugmentation for likely attributes.
2. Then use nc:PersonDocumentAssociation to associate a person with a document (in this case, a passport) with begin/end dates if needed.
3. It may be possible to eliminate scr:PassportType if all the attributes from scr:PassportAugmentation are carried forward to nc:PassportType.

Details of proposed change are attached

----- Comment #5 From [Christina Medlin](#) 2012-07-06 14:46 [reply] -----

Marked issue for upcoming review.

----- Comment #6 From [Christina Medlin](#) 2012-08-09 15:52 [reply] -----

The recommendation was discussed during the 8-9-12 telecon. The tiger team will review the proposal again to work out remaining issues.

First Last Prev Next No search results available [Search page](#) [Enter new issue](#)

Actions: [Home](#) | [New](#) | [Search](#) | issue # [Find](#) | [Reports](#) | [My Requests](#) | [Log out](#) test@test.org

Edit: [Prefs](#)

Saved Searches: [My Entries](#)

... Continued.

Comments #2, #3, #4 (no direct replies in this example)

Blue navigational banner menu at bottom of every page

7 NIEM Domain Update (DU) Process

Normally, a major release cycle will integrate published NIEM DUs identified by domains for incorporation into the release. At the start of NIEM 3.0 there is only one published DU (i.e., Children, Youth, Family Services - CYFS). So, while CYFS may use its DU as input, other domain inputs to NIEM 3.0 will be in the form of change requests (spreadsheets) or NIEM-conformant XSD.

Though DUs will not be necessary for NIEM 3.0, for completeness this section provides an outline of the DU process defined in the NIEM High Level Version Architecture (Reference: <http://reference.niem.gov/niem/specification/high-level-version-architecture/>). Domains can use this process after NIEM 3.0 (in particular, to publish domain changes that missed 3.0 deadlines). As stated earlier the DU process is part of continuous preparation for the next release and provides domains controlled independence for making changes between major and minor releases. (Reference: <http://reference.niem.gov/niem/specification/domain-update/>)

7.1 Domain evaluates the need to publish changes.

If a domain determines there is a need or requirement from its community for content changes, a domain may decide to publish a DU ahead of the next NIEM release. If content changes have a low priority, a domain should wait for the next NIEM release to integrate changes. A DU is an MPD that includes: DU schema, XML change log file, XML catalog file, conformance report, and optional documentation.

7.2 Domain gathers requirements for the DU.

Domain reviews requirements and determines whether to publish an incremental or a replacement DU.

1. If most changes are additive or limited in scope, a domain may decide to publish an incremental DU.
 - a. An *incremental* DU schema only defines new or modified data components. So, if a domain wanted to add/change 10% of its content, it should not duplicate the other 90% that did not change from the original schema.
 - b. An *incremental* DU schema will generally be used in IEPDs in addition to the original domain schema from the release.
2. If changes modify a lot of existing content, a domain will likely need to publish a *replacement* DU.
 - a. A *replacement* DU schema defines all of the components for the domain – new, modified, and unchanging ones. Previously existing components from the release may also be removed from the DU schema.
 - b. A *replacement* DU schema may have a lot of overlap and duplication with the original domain schema from the release.
 - c. A *replacement* DU schema is used in IEPDs instead of the original domain schema from the release.

7.3 Domain specifies changes in an approved input format.

1. NIEM-conformant XML schema (XSD) – Available tools: NIEM Model Editor, XML tools (e.g., XMLSpy, etc.).
2. Change request (CR) spreadsheet – Requires less knowledge of NIEM conformance rules – labeled columns abstract some of the details needed for NIEM schema conformance. (Reference: <http://reference.niem.gov/niem/resource/change-request/>)

7.4 Domain records each change represented by the DU in a NIEM XML change log.

If assistance is needed, release manager can generate a basic change log that captures differences between the original domain schema from the release and the DU. The domain may need to

review, refine if necessary, and add descriptive comments.

(Reference: <http://reference.niem.gov/niem/specification/changelog/>)

7.5 Iterative quality assurance (QA) process.

1. If changes are in XSD format, domain should run basic XML validation, check it with the NIEM Conformance Testing Assistant (ConTesA), and fix any errors identified.
2. Domain submits validated DU schema or CR spreadsheet to release manager.
3. Release manager runs additional automated QA checks and may return a list of QA or conformance issues to the domain.
4. Domain fixes issues, makes additional refinements as needed, and resubmits.
5. QA process continues until all checks are passed.

7.6 Release manager loads DU to staging server SSGT and generates DU schema.

1. Domain conducts final review on DU schema.
2. If needed, domain may make additional changes if needed and resubmit DU for another QA and conformance check.
3. Domain may use the SSGT staging server for testing (e.g., view content, generate subsets, and build small beta IEPDs).
4. The SSGT staging server may be taken offline and updated as needed; therefore, it is not guaranteed to be available continuously. The standard SSGT production server is not used for testing and review; and will only be updated when the final NIEM 3.0 is released.

7.7 Domain approves the DU schema and package.

Once the domain is satisfied with the DU, the release manager publishes the DU to publication area (Reference: <http://publication.niem.gov/niem/domains/>) immediately and updates the SSGT production server as soon as possible. An update to the NIEM SSGT production server can take several days if other SSGT changes are in the process of integration and testing.

8 NIEM Data Modeling Guide – Questions, Considerations, Decisions

Data modeling in NIEM is an attempt to represent real data exchange requirements in XML by balancing precision meaning with maximum reusability. It is rarely easy to reach consensus on one representation. There are usually many perspectives. These questions are designed to assist modelers to consider as many perspectives as possible. That said, it is not likely modelers can answer all questions in any given case; nor is it likely they can satisfy all potential conflicts that may be exposed by answers to these questions. That said, simple and practical is usually better, even if it is not an exact representation of the real world.

For each potential NIEM data component (type or property):

1. Does it belong in NIEM?
 - a. Is it required in NIEM information exchanges (IEPD) or messages? (scope)
 - b. What kinds of data exchanges use it or would use it?
 - c. Is it reusable in other messages?
 - d. Can it be used in composing IEPDs?
 - e. Does it overlap multiple requirement sources (models)?
 - f. What data models contain this component or one similar to it? (source/requirement)
2. Is it a characteristic or subpart of an existing (and the appropriate) NIEM type?
3. Is it too specific (not reusable)?
 - a. Is it general enough to include in NIEM Core? Should it be generalized?
 - b. Is it more specific to a Domain and therefore should be an extension to Core?
4. Does this property or type overlap with another? Not distinct?
 - a. Should it be joined with another component?
 - b. Should it be split into multiple components? (too broad)
 - c. Select one; remove the other(s)?
5. Is its name meaningful to NIEM?
 - a. Is it properly named?
 - b. Are the terms of its name meaningful??
6. Is its definition meaningful in NIEM?
 - a. Is it defined distinctly from other NIEM data components?
 - b. Will this component definition be clearly understood by others?
7. Is it modeled correctly (structurally correct; correspond well to the real-world)?
 - a. Does it have the right base-types and properties?
 - b. Is it missing common characteristic or subpart properties (simple/complex)?
 - c. Does it contain properties that should be deleted, replaced, or changed?
8. How is it related to other types (objects)?
9. How is it used? (as the primary context? in multiple contexts?)



<NIEM>

NIEM.gov